

How to add lessons to ShaderSchool

All lesson files are located in the subdirectory “lessons”. For each language there is another subdirectory. Currently there are two: “en” for English lessons, and “de” for German lessons.

Each lesson consists of a XML file and a directory containing all files required by the lesson. For example “Lesson 1: Introduction” is made up of the file “lessons/en/lesson1.xml” and all files in the directory “lessons/en/lesson1”. The directory contains the HTML file, shader sources, images and textures.

Every new lesson has to follow this standard. So first create a new directory for your lesson. The name does not matter, but better stick to the convention and call it “lesson#”, where # is replaced by your lesson number.

Next you have to create the HTML that contains the lesson text and the task definition. ShaderSchool uses the Qt HTML renderer, which has a weird opinion of what HTML should look like. You are advised to use a HTML file from an existing lesson as a template, this way all lessons share the same design and you avoid any Qt HTML headaches.

Create two shader files (vertex shader and pixel shader) that contain the solution to your task and two shader files that make up the starting point for the student.

Put all these files in the newly created directory. Make sure that you include all images that the HTML file needs as well as any textures needed by the shaders.

Finally you have to create a XML file along with the existing ones. This XML file is read by the application at startup and your lesson will be added to the corresponding menu entry.

Use the following as a template for your XML file:

```
<?xml version="1.0" encoding="ISO-8859-1" ?>
<lesson title="Lesson #: Your lesson title">
  <html file="lesson#\your_html_file.html"/>
  <vertexshader file="lesson#\your_vertexshader.vert"
    solution="lesson#\your_vertexshader_solution.vert"/>
  <pixelshader file="lesson#\your_pixelshader.frag"
    solution="lesson#\your_pixelshader_solution.frag"/>
  <gl>
    <!-- The background color of the GL window in the format "#RRGGBB" -->
    <background color="#808080"/>
    <!-- Choose one of the following objects to be displayed -->
    <object type="cube" size="0.5"/>
    <object type="teapot" radius="0.3"/>
    <object type="sphere" radius="0.4" stacks="20" slices="20"/>
    <object type="flag" stacks="12" slices="20"/>
    <!-- Delay in msec between each GL window refresh -->
    <update interval="1"/>
  </gl>
  <!-- Define the textures your shader needs and to which uniforms they belong -->
  <texture file="lesson#\texture_in_unit0.png" uniform="tex0"/>
  <texture file="lesson#\texture_in_unit1.png" uniform="tex1"/>
</lesson>
```